DOCUMENT RESUME

ED 318 756 TM 014 787

AUTHOR Kreft, Ita G. G.; Kim, Kyung-Sung

TITLE GENMOD, HLM, ML2 and VARCL, Four Statistical Packages for Hierarchical Linear Regression: Similarities and Differences.

SPONS AGENCY Netherlands Foundation for Educational Research, The Hague.

PUB DATE Apr 90

NOTE 30p.; Paper presented at the Annual Meeting of the American Educational Research Association (Boston, MA, April 16-20, 1990). Paper is part of "Comparing Four Different Statistical Packages for Hierarchical Linear Regression, GENMOD, HLM, ML2, and VARCL" by G. G. Kreft et al. (1990).

PUB TYPE Reports - Evaluative/Feasibility (142) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC02 Plus Postage.

DESCRIPTORS Algorithms; Comparative Analysis; *Computer Software; Computer Software Evaluation; *Estimation (Mathematics); Maximum Likelihood Statistics; *Regression (Statistics)

IDENTIFIERS GENMOD Computer Program; *Hierarchical Linear Modeling; HLM Computer Program; ML2 Computer Program; VARCL Computer Program

ABSTRACT

A detailed comparison of four computer programs for analyzing hierarchical linear models is presented. The programs are: VARCL; HLM; ML2; and GENMOD. All are compiled, stand-alone, and specialized. All use maximum likelihood (ML) estimation for decomposition of the variance into different parts; and in all cases, computing the ML estimates involves complex non-linear expressions in the parameters. In all cases, the equations must be solved by an iterative procedure. The major differences among the programs lie in the choice of the criterion to optimize and the choice of the algorithm to optimize the criterion. For each program, the design philosophy, implementation details, and routines are specified. Use of these programs is illustrated with data from four sets: (1) the Second International Mathematics Study, using 3,691 cases; (2) data from 1,290 students in the Netherlands (the GALO study); (3) data from 5,310 secondary students in the Netherlands (the VOCAL study); and (4) data from 96 students from junior high schools in Los Angeles (California). These illustrations reveal the applicability of software for hierarchical linear regression. Ten tables contrast the results obtained by the various computer programs. (SLD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☑ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
ITA G. G. KREFT
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ED318756

GENMOD, HLM, ML2 AND VARCL, FOUR STATISTICAL PACKAGES FOR HIERARCHICAL LINEAR REGRESSION.

SIMILARITIES AND DIFFERENCES

ITA G.G. KREFT *
CENTER FOR THE STUDY OF EVALUATION

KYUNG-SUNG KIM
GRADUATE SCHOOL OF EDUCATION

Discussion paper in the symposium "Recent Issues in Hierarchical Linear Modeling: Computer Programs, Philosophies and applications Compared".

AERA annual meeting
Boston, April 16-20, 1990

This article is part of a more extensive and more mathematical report, published as a CSE Technical Report at the Center for the Study of Evaluation, Moore Hall 145, UCLA, 405 Hilgard Avenue LA, CA 90024-9969. It can obtained by ordering report #310 under the title: "Comparing four different statistical packages for hierarchical linear regression, Genmod, HLM, ML2 and VARCL"authors: Ita G.G. Kreft, Jan de Leeuw and Kyung-Sung Kim (1990)

*My time dedicated to this research is made possible by a grant from the Netherlands Foundation of Educational Research, SVO project 7130

TM014787

BEST COPY AVAILABLE

1. INTRODUCTION.

1.1. Why new techniques ?

Evaluating the effectiveness of large-scale studies in education involves the analysis of *hierarchical data structures*. Educational data are often hierarchical because pupils are in schools, schools are in districts, districts are in counties, and counties are in states. In a large-scale research project we usually have information about two or more of the levels involved, for instance variables describing individuals (such as intelligence, school career, family background, and so on), variables describing the schools (school type, school in a special program or not, curricula offered, etc.), and perhaps variables describing districts or countries (the available resources, etc.). It is well known that analysis of these variables on any of these levels separately can be seriously misleading. For overviews we refer to Burstein (1980) and Kreft (1987). It is more satisfactory to construct models and techniques which simultaneously take information of all levels into account. But in order to be able to do this, some serious statistical problems have to be solved. Problems in hardware and software were unsolvable until recently. In the last few years, however, there have been a number of papers in the statistical and methodological literature which directly address the problem of analyzing variables measured at different levels of a hierarchy. We refer to Mason, Wong, and Entwisle (1985), Aitkin and Longford (1986), Goldstein (1986), Raudenbush and Bryk (1986), and De Leeuw and Kreft (1986). These investigators all work with basically the same model, known under several names as the *hierarchical linear model*, *random coefficient model*, or *Bayesian linear model*. All models deal with the problem of analyzing nested data collected under non-experimental conditions.

Mathematical models can serve the goal of describing relationships between variables of different levels. The model functions as a formal way of describing the empirical world, or as a formalized theory. Using these mathematical models means that we assume that the structure of the models is more or less in correspondence with that of the real world. Preference for one model over another is based on theoretical knowledge of the subject the researcher is dealing with and/or on the knowledge of the way the data are produced.

For instance the hierarchical linear model, the model we are discussing here, is preferable to the traditional models, sucl. as multiple regression and analysis of covariance (ANCOVA), when the assumptions of the latter are violated. As a result of this violation, we assume that estimators of the more realistic model are less biased and more efficient. Let us give an example with educational data, comparing the hierarchical model with a traditional fixed linear model such as ANCOVA. In ANCOVA the most obvious assumption violated is that of random sampling. This is clear, since in education we are dealing with data in which students are sampled from within schools and more often than not even these schools are not a random sample (but are stratified or otherwise sampled). Moreover, school populations do not consist of a random sample of the country's population of students, but are highly selective, causing a restriction of range for many important variables. Selection can take place on the dependent variable school success, since schools are in poor or rich neighborhoods, and the relation between school success and social economic status is obvious in many instances. In addition, when planning to do separate within school analyses, the samples within schools are frequently not big enough to make up for this non-normality of the data. As a result, the ordinary least squares (OLS) estimates of the traditional linear model provide confidence intervals that are too short and too often the null hypotheses will be rejected. This causes a lot of type I errors, and as a result contradictory research results.

The question we study in detail is how to fit separate models for separate schools in an elegant and parsimonious way. In order to do this we present a detailed comparison of four computer programs for analyzing hierarchical linear models. The programs we discuss are (a) VARCL, (b) HLM, (c) ML2, and (d) GENMOD. We have selected them on the basis of the following criteria. We restrict our attention to programs that are (a) compiled, (b) stand-alone, and (c) specialized. On the basis of (a) and (b) we exclude the routine by Stram, Laird and Ware (1986), which is written in SAS/MACRO and in SAS/MATRIX, and the routine by Kim (Kim and Kreft, 1989), which is written in GAUSS. Because of (c) we exclude general purpose mixed-model ANOVA routines, and because of (b), we do not discuss package modules such as BMDP/5V (Schluchter, 1987) or GLM and VARCOMP from the SAS statistics package. The routines of Jennrich and Schluchter (BMDP5V), and of Stram, Laird, and Ware (GGCMAOV), were written with longitudinal (growth curve, repeated measures) data in mind. They will be discussed and compared in a separate paper (Van der Leeden and De Leeuw, in preparation).

The four programs give their own answers, but basically their answers lead to the same conclusion. OLS produces biased and unstable estimates of the parameters and their precision, since the assumptions on which OLS is based are clearly violated. The specialized methods can produce more stable estimates. There are some quite subtle differences between the models that can be fitted by the programs, but generally there is a great deal these various approaches have in common. Most programs concentrate on two parts, or *levels* to decompose, although having more than two is possible (recently HLM3 and ML3 have been released in beta-form, VARCL already exists as VARCL3 and VARCL9). Using an educational example again, these levels may be the variance between students and the variance between schools. In this example, students are the first level and schools the second, but the levels could be defined differently. All four programs use *Maximum Likelihood* (ML) estimation for decomposition of the variance into different parts. In all cases computing the ML estimates involves complex nonlinear expressions in the parameters. In such cases the equations must be solved by an iterative procedure. The major difference between the programs is in the choice of the criterion to optimize, and the choice of the algorithm to optimize the criterion.

1.2. Models.

There has been quite a bit of confusion in the literature about the precise differences between the programs, the distinction between restricted and full ML, the precise definition of the algorithm, and the various reasons one can choose between programs and options. This may be due to the fact that the investigators have concentrated on making the software work, and getting application oriented papers published. It also may be due to the fact that some things are taken for granted, which are not obvious at all. Why do we use ML, why restricted ML instead of full ML and what is the EM algorithm ? Since restricted ML adds bias correction terms on each iteration to ML estimates, this amounts to treating the fixed coefficients as quantities which have uncertainty "built in", when computing the random parameters. ML, on the other hand, treats the fixed coefficients as known in the same situation. For small data sets the different estimation procedures may have different results. It is not clear for the user when to use ML and when to use Restricted ML (REML). This non clarification of the method used is a general omission of all four software packages, but especially clearly absent in ML2 , since the choice is here explicitly given between the two [note that REML is comparable with the RIGLS (Restricted Iterative General Least Squares) used in the English software ML2 and ML with IGLS also used in ML2). For literature on RIGLS we refer to Goldstein (1988)].

1.2.1. Multilevel Models.

We now define multilevel models in a quite general way. Suppose we have a hierarchy $\{\Pi_r\}$. The outcome variable $\underline{y}$ is of the most refined level s. In fact we assume that s = n, i.e. none of the $\underline{y}_i$ are *necessarily* equal (though they could very well be *accidentally* equal, think of the case in which the outcome is binary, as in passing or not passing an exam). For each level r we have m_r predictors of level r, all defined on $\mathcal{N}$. They are collected in the matrices $X^{(r)}$. Our first equation is

$$(1) \quad \underline{y}_i = \sum_{j=1}^{m_s} x_{ij}^{(s)} \underline{\beta}_{ij}^{(s-1)} + \underline{\varepsilon}_i^{(s)}$$

Thus, for example, columns of $X^{(s)}$ are predictors on the student level (and remember this does not exclude that there are school variables, county variables, and even constants among them), and the columns of $\underline{B}^{(s-1)}$ of are random coefficients on the school level (that is, if students i and k are in the same school then their beta's are equal). The disturbances $\underline{\varepsilon}^{(s)}$ in (1) have the usual structure. They are independent, identically distributed, centered, and they are independent of the random coefficients.

Now take the second step in the specification. For the elements of $\underline{B}^{(s-1)}$ we assume a model of the form

$$(2) \quad \underline{\beta}_{ij}^{(s-1)} = \sum_{l=1}^{m_{s-1}} x_{ijl}^{(s-1)} \underline{\beta}_{ijl}^{(s-2)} + \underline{\varepsilon}_{ij}^{(s-1)}$$

which means that the random regression coefficients of level s - 1 are linear combinations of fixed predictors of level s - 1, and new random regression coefficients of level s - 2 (plus a disturbance term of level s - 1). Observe that $X^{(s-1)}$ and $\underline{B}^{(s-2)}$ both have m_{s-1} columns. Disturbances $\underline{\varepsilon}^{(s-1)}$ are independent of previous disturbances, and of disturbances of the level s - 1 corresponding with different equivalence classes. They are identically distributed over equivalence classes. Of course they have zero expectations. We do not assume that they are independent within equivalence classes (for different variables j and l).

Anyway, it is clear how to continue from here to the next level. On level s (which is level n) we have nm_s random coefficients, which form m_s variables of level s - 1. Thus they take at most $m_s r_{s-1}$ different values. On the second level we have $nm_s m_{s-1}$ random coefficients, forming $m_s m_{s-1}$ variabes of level s - 2, that is taking at most $m_s m_{s-1} r_{s-2}$ values. And so on. In order to stop the sequence starting at (1) and (2), we usually assume that coefficients on the last (crudest) level are no longer random (have zero disturbances).

1.3 Algorithms

The four programs we discuss use three types of algorithms. One way to proceed is to think of the likelihood function as depending on two sets of parameters: the fixed regression coefficients η_j and the variance components Ω_j and σ_j. If the variance components were known, the regression coefficients could be estimated easily by weighted least squares. If the regression coefficients are known, the variance components can be estimated fairly simply as well. So the idea is to alternate these two minimizations iteratively.

The second type of algorithm, proposed for multilevel models by Longford (1987) and De Leeuw and Kreft (1986), is simply the classical method of scoring. This is the Newton-Raphson method, applied to the likelihood function with a convenient first-derivative approximation to the second derivatives. Full implementation of the Newton-Raphson method, for general mixed linear models, were discussed by Jennrich and Schluchter (1986) and by Lindstrom and Bates (1988).

The third type of algorithm is the EM algorithm of Dempster, Laird and Rubin (1977). It was applied to covariance component models by Dempster, Rubin, and Tsutakawa (1981), and to (longitudinal) multilevel linear models by Stram, Laird and Ware (1986). Compare also Jennrich and Schluchter (1986), Lindstrom and Bates (1988). Multilevel programs using the EM algorithm were written by Bryk, Raudenbusch, Seltzer, and Congdon (1988), and by Mason, Wong and Entwisle (1983). The EM algorithm is based on a clever bounding of the likelihood function with a more convenient minorization. In each iteration the minorization is maximized, and each of these steps increases the likelihood function as well.

Theoretically a comparison of the algorithms should be based on the fact that EM typically has (very) slow linear convergence, which is global, (i.e. which occurs from any starting point). Both weighted iterative least squares and scoring have fast linear convergence and, for models which fit very well, scoring will be almost quadratic. Both methods have global convergence as well. The Newton-Raphson method is truly quadratic, but it may diverge from the starting point when it is not appropriately chosen. In this paper we compare EM (used in GENMOD and HLM), scoring (used in VARCL), and weighted iterative least squares (used in ML2). From our results so far, it seems that in practice the results are less clear than the theory suggests. But precise results will also depend on the details of the parametrization and on the particular expression for the likelihood function that is used. In the following table algorithms define columns, criteria or loss functions define rows.

	EM	Scoring	IGLS
Full ML		VARCL	ML2
Restricted ML	GENMOD HLM		ML2

1.4 Limitations

The above multilevel techniques have various limitations, which restrict their applicability to educational research data. First, they are design-oriented, assuming fixed regressors on the individual level. In most evaluation studies the regressors are sampled in the same way as the dependent variables, and thus this assumption is not appropriate. The second limitation is that the techniques deal with a single dependent variable, predicted by a number of independent variables. In evaluation studies, and in school research in general, there is often more than one criterion to predict. One could apply the technique to each criterion separately, but this is not satisfactory because it ignores the relations between criteria. Thus a multivariate extension is needed, and more generally we would like to have the capability of fitting path models with multilevel techniques. A third limitation is that the algorithms, or at least some of them, cannot handle missing data and categorical dependent variables. A fourth one is the linearity of the existing techniques, and a fifth is the assumption of normality of the residuals. While these last two are also serious restrictions of generality, developing a technique which does not impose these restrictions involves a more far-reaching generalization of the existing methods, than developing multi-level path analysis techniques.

2. DESCRIPTIONS OF THE FOUR SOFTWARE PACKAGES.

2.1. GENMOD

A. Design Philosophy

GENMOD is written by Benjamin Hermalin at The Population Studies Center, University of Michigan, from instructions provided by George Y. Wong and William M. Mason. The program implements the general model proposed by Wong and Mason (1985), see also Mason, Wong and Entwistle (1984) for the general theory. It is a multilevel model, according to our definition, because it allows the user to specify the macro variables to be used as regressors in a macro regression in which the regressand is the micro coefficient or micro intercept.

GENMOD is developed to accommodate two broad classes of applications:

(1) comparative analysis,
(2) contextual analysis.

The contextual analysis is the usual one, as we find it in the other programs. Special is the comparative analysis. The presumption is that we have a different file for each context. These files may even have different formats. Moreover the micro data file for one context may also contain variables that are different. This characteristic of the program is very valuable in the field of demographics, where it was developed. As shown in the paper of Mason Wong and Entwistle (1984), countries may differ in their methods of birth control, in the way birth control clinics operate and/or to the extent they are available in the different countries. At the same time the background variables of the subjects (women) involved in the different countries may be defined differently. The latest version of the program, dated April 1989, provides the opportunity to read a simple dataset with a single micro file and a single associated format statement. We used this version in our comparisons of the programs as reported in paragraph 3.

B. Implementation details.

The program is written in Fortran77 and is currently compiled to run under the MS-DOS and MTS operating systems. File names must satisfy MS-DOS file naming conventions. There are three versions (GEN30, GEN40 and GEN50), which differ in the size of the real array storage that is been allocated ; 35,000, 45,000 and 55,000 elements of REAL*8 storage The distribution includes source code, however, which means that (at least in principle) any DOS or OS/2 user with a FORTRAN compiler can make his own version, with storage requirements adapted to his own environment. In practice this may still lead to some problems. We have tried to compile GENMOD with the Leahy FORTRAN compiler under DOS, with the VS-FORTRAN compiler under MVS, and with the Language Systems FORTRAN compiler under the MacOS, and none of these attempts have so far been successfull.

The memory available for array storage is the major constraint on the size and complexity of the model that can be handled. The array space required for a given setup is allocated dynamically from a single large array that has been defined in the source code. The three versions of the program differ only in the size of that array. The amount of array space required depends largely on an interaction between the number of contexts to be handled and the complexity of the model.

The memory requirements are greatest for the combination of a large number of contexts with a model that specifies fixed effect coefficients and OLS estimation for the start values. Under MS-DOS, the program reads and writes ASCII files only.

C. Routines.

The basic algorithm of GENMOD performs the following three steps:

(1) Constructs cross-product matrices and G-matrices from raw data.
(2) Performs ordinary least squares (OLS) estimation to obtain initial values.
(3) Performs EM iterations.

The user can activate the three steps by choice. So any of the three can be the starting point of the analysis. To activate step (1) the user must supply raw data files. To activate step (2) the user must supply a file with cross-product matrices. To activate step (3) the user must supply a file containing starting values for the micro and macro error variances and covariance.

As a result of the above options the program provides the user with two output files. One contains the statistical output, the other contains the information that can be used in subsequent analysis to activate step 2 or 3.

The maximum number of iterations is specified in the batch job. If convergence has not been achieved by the NUMIT-th iteration, the program will nevertheless stop, giving complete output as of the NUMIT-th iteration. The estimation procedure is REML. The EM algorithm used is based on equations developed by Mason and Wong.

2.2. HLM

A. Design Philosophy

The HLM, version 2.0, program has been written by Anthony Bryk, Steven Raudenbush, Mark Seltzer and Richard Congdon (1988). It is the most popular program in the USA, for several reasons: easy to use interactive interface, and a large number of special output (including different tests), which are absent in the other programs. The familiarity of the tests, such as T-tests and CHI squares tests, provides the user with a sense of trust in the results. Another explanation is the educational character of the manual. It provides a theoretical background for multilevel modeling and an abundance of references for more study.
HLM is developed to accommodate two broad classes of applications:

(1) contextual analysis
(2) growth curve analysis

B. Implementation details.

There are basically two versions of the program available. The first one is for workstations or mainframe computers with no real restrictions on the memory. The second one is an adaptation for the PC, and it requires 640 K memory limit of DOS. Both versions are written in FORTRAN 77, although the main program and some screen control functions have been written in C. This mixed language feature of HLM, plus the particular type of screen control used, make the program less than completely portable, although the computational routines are straightforward FORTRAN 77. The source of HLM 2.1 is no longer in the public domain, which means that the question of portability is not very interesting any more.

C. Routines.

Two routines are given in the manual: (1) an EM algorithm with an Aitkin accelerator, used as the core routine in the HLM program, and (2) a V-known routine. The latter is a more special algorithm. To generate starting values. HLM uses an ANOVA-type procedure. When the starting procedure of EM fails to produce a positive definite tau matrix the automatic fix-up routine, used in earlier versions, is replaced in the recent version (HLM 2.1) by the following options: (1) the decision to stop the analysis and to 'quit'. (2) setting all the diagonal elements to zero, (3). resetting these values by plugging in the users' own solution, or (4) trying the original solution: using the automatic fixup. The difference between the above solutions (except the first one, 'to quit') is in the number of iterations needed for estimates of tau and sigma squared.

The number of iterations is, as usual, optional and left to the decision of the user. The suggested number of iterations for exploratory analysis is 10. In most packages the advice is similar: up to 10 to 15 iterations. In paragraph 3 of this article we will show that for EM algorithms, this number is definitely too small to get results comparable to those of other programs.

2.3. ML2

A. Design philosophy.

ML2 is software for two-level analysis (ML3 for three level data has recently arrived) by Rabash, Prosser and Goldstein, based on Goldstein (1987) and part of a more general statistical package NANOSTAT. It is produced as part of the Multilevel Models Project, Institute of Education at the University of London. This project is funded by the Economic and Social Research Council of the United Kingdom to extend the theory of multilevel modeling, to study the practical application of the models to real data sets and to disseminate information about the theory and practice of this form of analysis.

B. Implementation details.

ML2 is only provided in binary form. It runs on DOS/OS2 computers. It needs 540 K of RAM. The remarkable aspect of the ML2 implementation is that the multilevel software is merged with the core of the general-purpose package NANOSTAT (Healy, 1987), which makes a whole set of data handling and data transformation operations available. NANOSTAT also provides descriptive statistics and high-resolution plots. If you want to use the graphics (that comes with the NANOSTAT package), you need a CGA, EGA, VGA, or Herculus card. We have had only minor problems with ML2 (in part III we shall report some bugs, which are presumably already corrected in the newest version). An important difference between ML2 and the other three programs is that the data are not first reduced to sufficient statistics, which are then kept in core memory. In ML2 the complete data matrix is read into core memory, which means that the restrictions on the size of the problem are more serious for ML2.

C. Routines.

An iterative generalized least squares (IGLS) algorithm provides estimates of model parameters, and maximum likelihood estimates when normality assumptions are met. ML2 can also compute unbiased or restricted RIGLS estimates, which are called REML estimates in other contexts. The user has the choice between IGLS, described by Goldstein (1986, 1987) and RIGLS described by

Goldstein (1989). IGLS is the default estimation method and is comparable with FEML used in the VARCL program by Longford, while RIGLS is comparable with REML as used in HLM by Bryk et al. The Restricted IGLS adds bias correction terms on each iteration to IGLS estimates. This amounts to treating the fixed coefficients as quantities which have uncertainty "built in", when computing the random parameters. IGLS, on the other hand, treats the fixed coefficients as known in the same situation. For small data sets the different estimation procedures may have considerably different results. It is not clear for the user when to use IGLS and when to use RGLS. This non clarification of the method used is a general omission of all four software packages, but especially clearly absent in ML2 , since the choice is here explicitly given between the two.

The first step in ML2's estimation process is to conduct an OLS regression using all the cases, ignoring grouping. The estimates for the fixed coefficients and for the residual variance become default starting values for the IGLS or RIGLS iterations. A user having some knowledge of estimates from modeling a particular data set may wish to start the iterations for a new model at a point thought to be closer than the default values to the new final estimates. See the paragraph "special options" below.

The number of iterations can range from 1 to 999. The default value is five iterations. This number is sufficient for reaching convergence when the conditions are favorable. Favorable means when the number of observations per unit is large enough to obtain stable estimations and the number of parameters to be estimated is small and the tolerance/convergence criterion is between 2 and 4. It is advisably to increase the number of iterations when the convergence is reached slowly, the amount of data is small and/or the number of parameters to be estimated is large.

When one of a pair of random coefficients has a variance of zero the covariance is set to zero. This default option can be switched off.

2.4. VARCL

A: Design Philosophy

VARCL was initiated by Aitkin and Longford (1986), and is produced and maintained by Longford. A variance component type of analysis, it can be used to analyze multilevel data. VARCL is a single purpose package, designed for the fitting of mixed linear models with nested random effects to data involving hierarchies of nesting. It consists of two independent computer programs, VARCL3 and VARCL9. VARCL3 is used for analysis of data with three or two levels of nesting, in which modeling of the variation of the within-area and within-group relationships is of interest. VARCL9 can be used for analysis of data with up to 9 levels of nesting, but it allows only simple structure of the random effects. There is no requirement of the balance of the nesting structure in either program. Since itt was developed as a variance component analysis, it provides the option to fit random slopes but does not allow interaction with random slopes and second (or third) level variables.

B. Implementation details.

Both programs are written in FORTRAN77 and require an interactive computing environment. It was originally written for VAX/VMS, but its has been ported successfully to PC/DOS and to Sun/Unix environments. VARCL3 is complex and has a more elaborate interface than VARCL9, but they are similar enough to warrant a single user's guide. The interface of VARCL combines interactive and batch features. The batch feature is the control file that contains declarations related to the data set such as the title, data file names (the data set may consist of several data files), formats, variable names, nesting structure, etc. Having this information available in a separate file makes the interactive session less tedious.

C. Routines.

For his VARCL program, Longford (1987) uses the Fisher scoring algorithm. The manual describes the algorithm in detail (quite unlike the black box approach in the ML2 manual). There are two possible complications that can occur during the iterations of the scoring algorithm. They are both dealt with explicitly and automatically by VARCL. If one of the estimated dispersion matrices becomes indefinite (has negative eigenvalues) during the iterations, the iterations are *damped*. Thus the program does not go all the way, but forces positive definiteness by interpolating. This may result in a considerable deterioration of the convergence behavior of the algorithm. More elegant solutions to this problem are possible, for instance updating a triangular factorization of the dispersions (Lindstrom and Bates, 1988). The programs prints the message that a *covariance adjustment* has taken place.

The second problem occurs when the information matrix, used in the scoring iterations, becomes singular. The offending parameter is then *aliased* (i.e. excluded from the model). Aliasing obviously improves the convergence, but results in fitting a different model. It is irreversible, i.e. once a parameter is aliased, it will not be unaliased and left free to vary anymore. It has been our experience with VARCL that aliasing occurs in situations with complicated models, in which the EM algorithms of GENMOD and HLM exhibit very slow convergence. Both in the case of aliasing, and in the case of covariance adjustment, the VARCL manual suggests fitting a smaller model.

2.5.GENERAL OVERVIEW OF THE FOUR PACKAGES

The usual restrictions imposed by the four programs are:
- all variables in the random part are included in the fixed part;
- all level one coefficients are random at level two (the full random coefficient model)
- the only explanatory variable whose coefficients could be considered to be random at level one is the intercept.

All four programs have ways to deal with the first two restrictions and leave room for variables that are not included in the fixed part to be random (HLM, ML 2), or to fit a mixed model (all programs), but the last restriction is only overcome in ML 2. On page 17 of the ML 2 manual an example is given in which a predictor of the first level has a random part at that same first level.

A test of significant improvement of fit between two models, a simple and a more complicated model for instance, is only provided in HLM. For the other packages, except for ML 2, it is easy

to calculate the test statistic by subtracting the deviances of two separate models. In GENMOD, however, the likelihood does not seem to be computed correctly. Deviances can especially be used effectively when using VARCL with its 'interactive fitting of several models' facility. The user can make alterations in the first fitted model, make further alterations, and so on, fitting consecutively any number of models in a single session. This feature is nice, because it allows one to find the most parsimonious model by comparing models against the best fitting, but unrestricted maximum model. By using the deviance from the maximal model of more than one solution, and comparing the difference with the degrees of freedom gained by estimating a smaller number of parameters, it is possible to make a choice between one model and another. Since the maximal model has the most parameters it fits, by definition, the best. The question is, How much better than a more useful (and easier to interpret) simpler model? The deviance statistic can be employed in a likelihood ratio test by comparing the deviance of two or more models. This difference between the two statistics has a chi-square distribution. The chi-square test of the significance of the differences in deviance, in relation to the gain in degrees of freedom, has to be calculated by the user in order to compare the fit of the different models. By doing so, it raises the awareness to the importance of the goodness of fit instead of an overemphasis on significant (separate) effects of coefficients and/or theoretical relevance of these effects. In HLM the deviance of the tested model from the 'full model' (the name of the maximum model) with the associated degrees of freedom and chi-square statistic are part of the output. Because the maximum model can be followed by any number of different models in VARCL, and as a result the number of comparisons can be extensive, it is impossible to give those statistics in the output. It is unfortunate that the VARCL manual does not provide guidelines for how to use the deviances in testing the differences in goodness of fit of the different models fitted in one session.

The number of iterations is variable in all packages and left to the decision of the user. The suggested number in most packages is from 10 to 15 iterations. We have shown that for EM algorithms this number is definitely too small to get results comparable to those of other programs. Comparing HLM and GENMOD with ML 2 and VARCL in Table 2a,b and 4a,b for instance shows this clearly. The latter two packages, with fast linear or superlinear convergence, stay within the limit of 15 iterations, while the other two exceed that number considerably in order to reach the same convergence criterion. In our examples more complicated models (more complicated than a model with only a random intercept and no second level variables) need quite a lot more iterations in the packages that use the EM algorithm (GENMOD and HLM) than in the ones that use scoring (VARCL) or weighted least squares (ML 2).

The (default) output given by the four packages differ from one page to several and from many parameters and significance tests to only the essentials. ML 2 gives as default only the fixed and random parameters with the respective standard errors and the number of iterations. This may be nice for new users who are not confronted with an abundance of output, which makes the relevant results hard to find. The more experienced user needs more output, which can be provided on request, such as OLS estimates and residuals. In this respect GENMOD resembles ML 2 and both differ a lot from VARCL and HLM.

T-test values for parameters are only provided in HLM. This is not unusual for this type of software, since separate t-tests do not take the total analysis into account, which may lead to type-I errors when making interpretations from single t-tests. Beside the tests are more heavily dependent on statistical assumptions, such as normal distributions, than parameter estimates are. Compared to the other packages we found that HLM provides a maximum of information in order to minimize the effort for the user. An example is that next to the usual estimates of the fixed part of the coefficients (gamma's) and their standard errors, the t-statistics and p-values are also given. In contrast: the other three programs provide only the first two and leave it to the user to calculate the significance tests when it is a necessary and/or the responsible thing to do. The overall test (test of

differences between deviances for the goodness of fit) may be a more reliable one here.

Packages differ in the way they handle the raw data set, especially with regard to centering, a much discussed issue lately (Raudenbush, 1989a,b, Longford, 1989b, Plewis, 1989). One of the options is to center variables around the group mean or grand mean. An option offered in the ML 2 package is the choice to center variables used in the fixed and random part around the grand mean in either, both or none of the parts with the options FMEA (fixed part) and RMEA (random part). The reason for centering, as given in the ML 2 manual (page 14), is that centering sometimes facilitates interpretation, but that it is mainly used as a way to improve the numerical performance of the estimation algorithm. In VARCL variables are replaced by their centered version to make computation easier. Here the deviation is a deviation of the grand mean and not a deviation of the group mean. No options are available to change this default centering in order to use raw scores. But since the outcome is reparametrized in terms of the original data this leads to the same outcomes as in programs that use raw data (as is shown in the comparisons in Chapter 3). This also leads to the conclusion that centering in VARCL is of no consequence. In HLM the authors have built in a question in their interface: "Do you want to center variables?", if the answer is "yes" the names of the variables to be centered have to be introduced. To center none, all, one or more variables is a choice left to the user. This is however centering around the group mean. There is no option in HLM to center from the overall mean.and has to be done before starting HLM. The manual explains that this type of centering may be advisable in order to make interpretation of the intercept easier.

Information about the convergence process and the use of this information is quite different in the four programs. It is possible in the program ML 2 to interrupt during the run and *freeze* the estimation of some individual parameters. This is helpful in situations where some parameter estimations seem to converge slowly because the estimated variance of those random parameters are close to zero. The freezing of the estimates speeds up the convergence. In the ML2 package the user is informed during the run of slow converging parameters. A somewhat similar concern is formulated in the manual of HLM. But the user is not allowed to take action during the run. In HLM, the user is informed afterwards by looking at the chi-square table and the deviance statistics of the run. By setting the residual parameter variance at zero in the next run for those variables that slowly converged, the same effect is reached in HLM as is reached in ML 2 by *freezing* during the run. These boundaries are handled in various ways. Programs using the EM method need no special provisions to deal with boundary constraints. Parameters can never get outside, variances are never negative, and matrices are never exactly singular. Nevertheless EM methods converging to boundary points generally have sublinear convergence (Horng, 1987). It is not entirely clear that this boundary is treated efficiently in the other packages (ML 2 and VARCL). What happens if a matrix is almost singular? Will the parameter be set to zero (to save time), or will the iterative process merely slow down. Differences between programs (or versions of the same program) may result in different solutions (see Chapter 3).

Hypothesis tests are possible within some packages. In ML 2 the residuals can be used in two different ways. First the residuals may be, and often are, used directly to provide estimates of the unknown macro effects as is explained in Aitkin and Longford (1986). ML 2 will provide estimates of the standard errors of these estimates, so that confidence intervals may be examined. When sample size is large, the estimates will have an approximately normal distribution and as a result confidence intervals can be estimated in a reliable way. Secondly, for hypothesis testing ML 2 can calculate simultaneous confidence intervals for a set of contrasts that you specify as well as a confidence interval for each contrast considered individually. This makes it possible to test complex hypothesis about several elements of the matrix gamma. By using a contrast matrix (equivalent to the matrices used in the Scheffé test for effects in ANOVA), several combinations of effects can be tested against each other without enlarging the alpha level. In HLM hypothesis

testing is also offered as part of the program, but in a slightly different way, basically with univariate and multivariate t testing.

The use of interaction variables, such as the interaction of a (fixed) macro level variable with a random micro level variable, are quite common in multilevel data analysis. But not all packages provide equal opportunities to do so. It is not a problem to use these type of interactions in theGENMOD and HLM programs, but less obvious in ML 2 and almost forbidding in VARCL. In the ML 2 manual this is emphasized in the first example. It is clearly indicated that interaction variables (interactions between the first and second levels) have to be created before the analysis starts. It can easily be done in 'the specification of the model'-phase by using the command MULT, followed by the two names of the variables and a position for the newly created variable (in the format statement). In the VARCL manual, no mention is made of the fitting of interactions between random first level and fixed second level variables. Basically VARCL does not fit random coefficient models, but variance component models. It allows for random slopes and for an interaction between group-level characteristics and intercepts, and not between slopes and group level variables. This means that the program does not create variables during the session, the way HLM and GENMOD do, nor does it allow the researcher to make these interactions as ML 2 does. An earlier version of VARCL did allow the user to make interactions, but this feature has been removed from later versions (also removed was a module to include various transformations of the variables). It is however possible to fit a model with interactions using VARCL, by fooling the program and use an input matrix with specially created interaction variables. By treating these interactions as fixed micro level variables in the model, the same results can be obtained as in the other packages, as will be shown Chapter 3.

Some packages allow dichotomous dependent variables. A specialty offered in ML2 is the opportunity to model a simple multilevel logit and log-linear model. This allows the researcher to analyze survey data with proportions or binary variables as the dependent variable. A simple multilevel logit and log-linear model is described in the manual . This feature is shared with Longford's program VARCL, who also includes an option for the error distributions to be either logistic, logarithmic or reciprocal. We understand that there is also a version of GENMOD to deal with logistic multilevel models, but we have not seen it.

	GENMOD	HLM	M L2	VARCL3
availibility	shareware, from author; source included	commercial, from software house	shareware, from author	shareware, from author; source available
literature references in manual to basic literature or applications of the hierarchical linear model	few, mainly to theoretical or basic work	many , mainly to applied work	many, mainly to applied work	few, mainly to theoretical or basic work
ease of use of the program and manual	the manual is difficult to read and the program set up complicated	the manual is clear and the program easy to use. No program set-up, but answers to questions.	the manual is clear, but it is hard to find the program set-up for a particular problem, due to the many choices	the manual is not easy but the program is easy to use. No program set-up, but an interactive answer and question system.
estimation procedures	restricted maximum Likelihood REML	restricted maximum likelihood REML	Generalized Least Squares, restricted or unrestricted IGLS and RIGLS	Full Information Maximum Likelihood FIML
possibilities within the package for data manipulation before or after the modelling stage	none, the data has to be prepared in advance	a lot; before, within, and after; see the manual	a lot; data exploration and preparation before and after modelling: plots of residuals etc.	none, the data has to be fully prepared in advance
Interface	batch job	fully interactive. The program asks questions, the user answers.	interactive, user gives a statement, programs answers	batchjob for data declaration. The run stage is fully interactive: the program asks the user answers
complication of the preparation of the dataset	Identification for both levels. Data have to be hierarchically ordered	identification and hierarchical ordering of the data	no preparation, all the necessary preparation (interactions and ID) can be done within the package	preparations of the datset can be complicated; necessary interactions prepared in advance
default output	in the batch job many parameters are available to ask for a lot or only the essential parameters	default output is a lot, but it can be expanded with more tests	output can be regulated from very essential to very much, by way of the commands used in the interactive stage	output is the whole history. Only the posterior means and the iteration history can be surpressed
Specialities and/or unique features	allows variation over context of residual variance. Allows different models over different contexts	hypothesis testing (shared with GENMOD) The estimation of reliability coefficients	The choice between restricted and unrestricted estimation. Proportions as the dependent variables.	Three levels and up till nine levels for simple models. Allows for proportions as the dependent variable.

	GENMOD	HLM	M L2	VARCL3
options for weighting	no	yes	no	yes
variance-covariance adjustments	not necessary, the EM algorithm	not necessary: the EM-algorithm	???	aliasing and covariance adjustments.
small data sets (# observations within groups less than # variables)	no	yes	yes	yes
performance	very slow convergence	faulty singularity test	fails for some small datasets in RGLS	adjusts covariances
documentation	not good	good	good	average
ease of learning	hard	very easy	very hard	easy
ease of use	moderate	easy	very easy	easy
error handling	good	moderate	moderate	moderate
speed	slow	fast/slow	not fast	very fast

3. COMPARISON OF THE PROGRAMS

3.1 Introduction

In this section of the paper we compare the various programs. The comparisons are far from complete, for various reasons. We briefly discuss those reasons here, describe what we intend to do about them, and give our further plans.

In the first place some programs are not finished products (although ML2 and HLM now seem to be approaching this stage). During our experiments with the program we discovered bugs in all four programs, and these bugs were (or will be) corrected by the authors after we let them know about them. We think that all four programs may still contain bugs of some sort, because they give strange output under some circumstances. As far as ML2 is concerned, we do not have the source code for this program, and we cannot really experiment with it in the same sense as with the other programs. It is difficult to decide which version to use. While we were printing the first version of this report, we received a new version of ML2, which was about three times faster as the orginal version we used. During the first and second version of this report HLM was updated to version 2.1, and VARCL was also modified. This is somewhat frustrating,in attempting to compare these four programs. We have tried to incorporate changes and bug fixes made by the authors as much as possible, but at some point we obviously had to stop. Changes reported after November 15, 1989, have not been incorporated in our comparisons.

Secondly comparisons are difficult, because the programs are different in various unfortunate aspects (at least for our purposes). ML2 does not write out the value of the likelihood function or the deviance, HLM writes the value of the restricted log-likelihood, VARCL writes the value of the unrestricted deviance, and GENMOD writes out both values (but minimizes the first one). For large examples GENMOD does not give sufficient precision in the output to compare values of the likelihood function with those of other programs (because the authors want to have the output for each iteration on a single 80 column line). The stopping criteria for each program are very different. ML2 and VARCL have fast linear convergence, which is actually close to superlinear if the model fits well. GENMOD and HLM typically have slow linear convergence, and the default stopping criteria for GENMOD are much more conservative than those of HLM. Thus, comparions of convergence should really be in terms of the likelihood function, but we have already seen that this leads to unexpected difficulties.

Thirdly programs may have restrictions, which may not have been intended by the authors. HLM refuses to perform at least some functions if a within-group cross product matrix is singular. GENMOD, in its original versions, had different individual-level error variances for each group, and could consequently not be compared with the other programs because of this. In the latest version the option of restricting all these error variances to be equal has been added. VARCL uses aliasing and covariance substitution in case of singularities, which is perhaps a good idea, but which again makes comparisons very difficult.

Our comparisons are on AT-type machines running DOS. Comparing the same program which fits the same model on the same data on different machines seems simple, but is actually quite complicated. In the first place not all programs run on all machines. Secondly complicated interactions between programs and machines make time comparisons very difficult.

3.2 Data, Models, and Results

We use four data sets, SIMS, GALO,VOCAL and Webb, which are described briefly before we give the results of the analyses.

3.2.1 SIMS

The SIMS (Second International Mathematics Study) data is a national sample of United States 8th grade students mathematics achievement tests conducted by IEA (the International Association for the Evaluation of Educational achievement) in 1981-1982. We have 3691 cases out of approximately 7500 extracted for this study. There are 190 school classes. Only two student level variables, the sum of PRETEST core items and the GAIN score (difference between POSTTOT and PRETOT), are used. The second level variable is Opportunity to Learn (OTL).

Thus the within-group model is

$$(\underline{GAIN})_i = \beta_{0j} + \beta_{1j}(PRETOT)_i + \varepsilon_i,$$

and the between-group model is

$$\beta_{0j} = \gamma_{00} + \gamma_{01}(OTL)_j + \delta_{0j},$$

$$\beta_{1j} = \gamma_{10} + \gamma_{11}(OTL)_j + \delta_{1j}.$$

The first version of the model has $\delta_{1j} = 0$ and $\gamma_{01} = 0$ (a random intercept model), the second version has both a random intercept and a random slope (and they are correlated), but still no second level variable (so $\gamma_{01} = 0$). The third model has, in addition, the macro variable OTL.

Table 1a: SIMS data, random intercept model, no macro variables

	GENMOD	HLM	ML2 (R)	ML2 (F)	VARCL
FixedPart					
(gamma)00	7.027	7.027	7.028	7.024	7.024
(gamma)10	-0.192	-0.192	-0.192	-0.1917	-0.1916
RandomPart					
(sigma)	22.52	22.52	22.52	22.51	22.51
(omega)00	9.443	9.443	9.442	9.374	9.374
Additional					
TotalIter	9	5	5	5	7
TimeIter	220	9	50	38	8
Likelihood/ deviance		22306.6			22380.7

Table 1a shows that the outcomes of comparable programs are the same up to two decimals. Remember that the convergence is not in all cases comparable, because the programs use sometimes different stopping criteria. Another difference is that the first three programs use RML

method, while the last two use FML. This is also the difference between ML2(R) and ML2(F), since ML2 offers a choice between the two estimation procedures. The difference in the solutions, which the different estimation methods produce (RML or FML) is obvious in all tables in this chapte. The difference is more pronounced in small data sets than in large ones, and also more in complicated models with random slopes than in simple models with only a random intercept (see Table 1c in this paragraph). By inspecting the different tables we see this clearly demonstrated. Compare the solutions for RML and FML of the larger SIMS data set with the smaller GALO data reported in the next paragraph. The data sets are respectively 3691 students in 190 schools and 1290 students in 37 schools. Compare also the solutions for the simple model in Table 1a versus the more complicated model in Table 1b in this section and between Table 2a and Table 2b in the next section.

The results of fitting these three models are in Tables 1a, 1b. and 1c. The rules for interpreting the these tables (and the other ones in this chapter) are simple. The easiest thing is to refer to the formulas for the corresponding model, which employ the same notation. The fixed regression coefficients (gammas) are given first, with (gamma)ij indicating the effect of macro variable j on micro variable i. Thus (gamma)00 is the fixed slope, (gamma)i0 is the effect of the macro intercept on the micro variable i (i.e. th[illegible]xed regression coefficient of variable i), and (gamma)0j is the effect of macro variable j on the micro intercept (i.e. the fixed regression coefficient of the macro variable j). In the same way (gamma)ij with both i and j not equal to zero is an interactive variable, the product of micro variable i and macro variable j. The second part of the tables then give the random components. We use (sigma) for the variance of the first level disturbance. The variances and covariances of the random part of the regression coefficients are the (omegas). The (omega)ii are variances of the random slopes, with (omega)00 the variance of the random intercept. Here (omega)ij is the covariance between the disturbances of the regression coefficients of variables i and j. Thus (omega)i0 is the covariance between the random components of the intercept and the slope of variable i.

In the last part of the table we give some additional information. TotalIter is the total number of iterations, while TimeIter is the time per iteration. Of course the product of the two is the total time spent in the iterative process. The total number of iterations depends, obviously, on the precision we impose. Our general rule here is to use the defaults, adjusted in such a way that the programs seem to converge with about equal precision. But generally it is difficult to do this in a completely satisfactory way, for the reasons referred to above. GENMOD has two stopping criteria: one at the absolute change and one at the relative change in the parameters. The others have as a stop criterion the absolute change in the parameter estimates. Only HLM has another stop criterion, it uses the (restricted) likelihood function. In some of the tables, but not in all, we also give values of the likelihood function or the deviance at the optimum.

Tables 1a, 1b and 1c are computed on a IBM PC/AT (with 80286/6Mhz CPU). The 286, although relatively the slowest of this type of machines, is used throughout this chapter, since this is the most common machine available to researchers. The time comparisons also include compiler and overlay effects.

In Table 1c, the most complicated model with a macro variable included, we see more difference in estimators over packages than in the other two models. All fixed parts are different over packages, but especially between the RML and FML ones (GENMOD, HLM, ML2(R) versus ML2 (F) and (VARCL).

Table 1b: SIMS data, random slope model, no macro variable

	GENMOD	HLM	ML2 (R)	ML2 (F)	VARCL
FixedPart					
(gamma)00	7.060	7.060	7.060	7.055	7.0553
(gamma)10	-0.186	-0.186	-0.186	-0.1857	-0.1858
RandomPart					
(sigma)	22.23	22.23	22.24	22.24	22.240
(omega)00	14.52	14.53	14.491	14.36	14.329
(omega)11	0.009	0.009	0.009	0.0088	0.00885
(omega)01	-.2342	-.2370	-0.2340	-0.2297	-0.229
Additional					
TotaIter	189	76	10	10	14
TimeIter	480	16	180	142	11
deviance		22382.4			22373.11

Table 1c: SIMS data, random slope model with OTL as macro variable

	GENMOD	HLM	ML2 (R)	ML2 (F)	VARCL
FixedPart					
(gamma)00	0.0627383	0.069168	0.06191	0.03242	0.039131
(gamma)01	0.234197	0.234021	0.2342	0.2349	0.234701
(gamma)10	-0.228330	-0.229388	-0.2282	-0.2236	-0.224472
(gamma)11	0.0008590	0.000891	0.0008554	0.0007268	0.000751
RandomPart					
(sigma)	22.1318	22.1265	22.13	22.14	22.139398
(omega)00	12.65	12.68382	12.64	12.38	12.362
(omega)11	0.0119	0.01141	0.01117	0.01046	0.010
(omega)10	-0.2302	-0.23294	-0.2300	-0.2205	-0.220
Additional					
TotaIter	145	59	10	10	13
TimeIter*	515	25	242	165	18
deviance		22367.8			22340.7

If we compare times over packages for the 286 machine we see that VARCL and HLM are clearly

the two fastest programs for simple models (with only a random intercept) as is shown in Table 1a (and Tables 2a and 4a in the next section). The time needed to reach the convergence criterion in the more complicated models with random slopes shown in Table 1b (and Table 2b and Table 4b in the next section), is many times larger (from four times more for GENMOD up till fourty times more for HLM). VARCL is still by far the fastest, and GENMOD by far the slowest, but the fast HLM program changes to a fairly slow one, slower than ML2(R) in most instances. The faster programs when fitting more complicated models are now the two that use FML and a faster algorithm, i.e. VARCL and ML2(F).

3.2.2 GALO

The data consist of 1290 students in 37 schools in the city of Groningen, see De Leeuw and Kreft (1986) or Kreft and De Leeuw (1989). For each pupil the individual level independent variables we used were gender (SEX) , IQ, and occupational level of the father (SES). The dependent variable ADV was teachers advice on the most appropriate form of secondary education (scaled as a continuous variable). There is no macro variable.
The within-group model is

$$(\underline{ADV})_i = \beta_{0j} + \beta_{1j}(SEX)_i + \beta_{2j}(IQ)_i + \beta_{3j}(SES)_i + \underline{\varepsilon}_i,$$

and the between-group model

$$\beta_{0j} = \gamma_{00} + \underline{\delta}_{0j},$$
$$\beta_{1j} = \gamma_{10} + \underline{\delta}_{1j},$$
$$\beta_{2j} = \gamma_{20} + \underline{\delta}_{2j},$$
$$\beta_{3j} = \gamma_{30} + \underline{\delta}_{3j}.$$

The basic results are in Table 2a for the random intercept and in Table 2b for the random slope. These tables show more discrepancies in the solutions between the different programs (R or F) than we noticed before in the first set of tables. The descrepancies are more apparent in Table 2b, where the more complicated model with random slopes is fitted. But the deviant behavior of HLM with respect to the intercept in Table 2a is an artifact of our own data manipulation. In order to get HLM started we centered all variables around the grand mean (centering of the independent variables only would have also done the trick, as we will discuss later). Using the raw data set was not possible in this case and ended in an error message which contained the following error message:

> At least some subset of the units must have sufficient data to permit OLS estimation within groups to generate starting values. One of the following problems or combination of problems has occurred in every unit:
> 1.No variation in the outcome measure;
> 2.No variation in one or more of the within-unit explanatory variables;
> 3.A singularity in within-unit data matrix for each unit.
> We suggest that you carefully examine for each group the sums of squares and cross products among the within unit variables. If these appear OK, try centering each of the within-unit explanatory variables around the grand mean. Since a condition check on X'X is included, near singular but technically invertible matrices are excluded from the starting value routine. Centering around the grand mean should solve this problem.

Table 2a: GALO data, random intercept model

	GENMOD	HLM*	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00	-4.6738	-0.0325	-4.6740	-4.6770	-4.6766
(gamma)10	-0.0901	-0.0901	-0.0901	-0.0903	-0.0903
(gamma)20	0.0804	0.0804	0.0804	0.0805	0.0805
(gamma)30	0.1487	0.1486	0.1488	0.1489	0.1489
RandomPart					
(omega)	0.9074	0.9075	0.9075	0.9053	0.9053
(omega)00	0.0489	0.0488	0.0488	0.0466	0.0466
Additional					
TotaIter	11	11	3	4	5
TimeIter	12	2.5	34	19	3
deviance		3601.77			3569.61

* For HLM all variables were centered around their grand means.

Table 2b: GALO data, random slope model, no macro variables

	GENMOD	HLM*	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00	-4.5393	-4.5361	-4.5290	-4.5280	-4.6195
(gamma)10	-0.0600	-0.0605	-0.0592	-0.0605	-0.0686
(gamma)20	0.0783	7.8263*	0.0781	0.0781	0.0793
(gamma)30	0.1497	0.1499	0.1519	0.1520	0.1497
RandomPart					
(sigma)	0.8786	0.8793	0.8824	0.8818	0.8846
(omega)00	0.7391	0.7206	0.6569	0.5768	1.1230
(omega)11	0.0422	0.0402	0.0376	0.0339	0.0280
(omega)22	0.0002	1.5321*	0.0001	0.0001	0.0000
(omega)33	0.0011	0.0009	0.0000	0.0000	0.0000
Additional					
TotaIter	214	772	13	15	13
TimeIter	75	11	217	167	11
deviance		3578.36			3559.03

* Coefficients are based upon IQ/100.

According to this message all 37 groups were deemed to have singular cross-product matrices, and no data are left to start the iterations. The authors suggest that transforming predictors to deviations from the mean may solve the problem. It indeed solved the problem, because HLM runs nicely after this transformation is performed. Centering of all the variables including the dependent one did give us the solution as reported in Table 2a. Since centering in this case clearly violates the invariance considerations (see CSE report, Kreft et al., 1990, Chapter 1), we looked into it somewhat deeper. The results are given in Table 3. The problem, we found out, is not the data set, but the conditioning test used by HLM. The authors are aware of the problem and will correct it in a subsequent release.

Table 3: GALO data, HLM results with different scalings of IQ

	IQ/10	IQ/100	IQDev	IndVarDev	AllVarDev
FixedPart					
(gamma)00	-4.5390	-4.5420	3.5190	4.0660	-.0682
(gamma)10	-.0604	-.0599	-.0611	-.0604	-.0603
(gamma)20	.7830	7.831	.0783	.0783	.0783
(gamma)30	.1496	.1497	.1493	.1496	.1496
RandomPart					
(sigma)	.8786	.8782	.8783	.8785	.8785
(omega)00	.7551	.7676	.1859	.1328	.0532
(omega)11	.0411	.0424	.0421	.0411	.0415
(omega)22	.0155	1.559	.0002	.0002	.0002
(omega)33	.0012	.0011	.0010	.0012	.0012
Additional					
TotIter*	101	99	116	103	98
Groups	36	37	36	36	36
Correlation**	-.9846	-.9833	-.9355	.4191	.4168
Deviance	3583.20	3578.60	3587.79	3587.81	3587.81

* Stopping criterion is 0.00005.
** Corr is correlation between St(BASE) and S2(IQ).

Looking at the different solutions given by HLM in Table 3, the most visible one is the change of the intercept from negative -4.53 to positive +4.06 and the change of the correlation between the intercept (base) and the slope of IQ, from negative -.9846 to positive +.4168. A lower correlation often means a faster convergence and a more precise estimate of the parameters. For the same reason, the VARCL starts its calculations with deviation scores, but recalculates the estimates back to the original values for raw data scores after convergence. Since HLM has chosen to start with the raw data, the user is sometimes forced to center the scores in order to obtain a solution. But the reason for using deviation scores instead of raw scores in our case is more related to other factors. The values of the variables we used have very unequal ranges, since IQ has a range of 60 to 130, while the other two variables (SEX and SES) have a much smaller range. In order to obtain a solution for our GALO example, it was sufficient to make the range of the variable IQ smaller and thus more equal to the scale of the other variables (SEX with two and SES with six categories).

The results of different manipulations are shown in Table 3. The first column of this table shows a solution with IQ divided by 10, the second with IQ divided by 100, the third with IQ in deviation from the grand mean and the fourth with SES, SEX and IQ in deviations from the grand mean. In the last column all variables are put in deviation of their respective grand means (as we did in Table 2a), and as a result the intercept is close to zero. In the first two columns of Table 3 the interpretation of the intercept is equal. It is the value for the dependent variable (advice) when the values of the independent variables (SES and IQ and sex) are zero. Since we have no situation in which SES and IQ are zero, this is a non-existing situation. Different is already the interpretation of the intercept in column three: when a student has a zero SEX (male) score, a mean IQ and a zero SES (still an unrealistic assumption) the predicted advice will be 3.5190 (the value of the intercept). The fourth column predicts a value of 4.0660 for advice for a student scoring average on all three independent variables. The changing values for the slope of IQ in Table 3 over solutions, (gamma)20 and (omega)22, are due to the different scalings of IQ. The changes in the variance of the intercept, omega(00), and the change of the correlation in the same table over solutions can easily be proven to be the result of using deviation scores instead of raw scores. All these changes do not make conclusions based on one of these solutions different from another.

A reason why researchers may prefer centering in some form is the ease of interpretation. Some other reasons for centering (around the group mean in that case) are discussed by Raudenbush (1989a,b), Longford (1989b), Plewis (1989). The centering issues will not be discussed but we will show the differences in interpretation of the intercep.

3.2.3 VOCAL

The individuals are 5310 students in secondary education in the city of Amsterdam in 1975. They are from 70 different schools. Individual level predictors are SEX and CITO, where CITO is a school attainment test at the beginning of secondary education. The dependent variable is CAREER, which is a scale based on a multiple correspondence analysis of the complete school career. For details we refer to Kreft (1987). A school-level predictor is TYP (which takes five different values, the five major types of secondary education in The Netherlands). Although this is not entirely appropriate TYP has not been used here as a categorical variable, but as a continuous variable.

The model we use in this study is

$$\underline{(\text{CAREER})}_i = \beta_{0j} + \beta_{1j}(\text{SEX})_i + \beta_{2j}(\text{CITO})_i + \varepsilon_i,$$

and the full between-group model is

$$\beta_{0j} = \gamma_{00} + \gamma_{01}(\text{TYP})_j + \delta_{0j},$$
$$\beta_{1j} = \gamma_{10} + \gamma_{11}(\text{TYP})_j + \delta_{1j},$$
$$\beta_{2j} = \gamma_{20} + \gamma_{21}(\text{TYP})_j + \delta_{2j}.$$

As usual, we fit a random intercept version, which has

$$\beta_{0j} = \gamma_{00} + \gamma_{01}(\text{TYP})_j + \delta_{0j},$$
$$\beta_{1j} = \gamma_{10} + \delta_{1j},$$
$$\beta_{2j} = \gamma_{20} + \delta_{2j}.$$

and a random slope version. with

$$\beta_{0j} = \gamma_{00} + \gamma_{01}(TYP)_j + \delta_{0j},$$

$$\beta_{1j} = \gamma_{10},$$

$$\beta_{2j} = \gamma_{20} + \gamma_{21}(TYP)_j + \delta_{2j}.$$

Results are given in Tables 4a and 4b.

Table 4a: VOCAL data, random intercept model

	GENMOD	HLM	ML2 (R)	ML2 (NR)	VARCL
FixedPart					
(gamma)00	1.2360	1.2360	1.2360	1.2340	1.2326
(gamma)01	-.2742	-.2743	-.2742	-.2731	-.2727
(gamma)10	-.0798	-.0797	-.0797	-.0797	-.0796
(gamma)20	-.3229	-.3229	-.3229	-.3232	-.3231
RandomPart					
(sigma)	0.3995	0.3995	0.3995	0.3995	0.3995
(omega)00	0.3484	0.3480	0.3483	0.3392	0.3563
(omega)10	0.0071	0.0070	0.0071	0.0068	0.0070
(omega)20	0.0962	0.0962	0.0962	0.0946	0.0960
Additional					
TotalIter	109	101	17	12	11
TimeIter	145	13	567	413	12
Deviance		10628			10593

It is clear from the Tables 4a and 4b that the packages give very similar results, for fixed as well as the random part, for the first level as well as for the interaction coefficients. The main difference is again between RLM and FML and between VARCL and the rest, but the differences are small. We expected that much, since the data are well conditioned and a large number of observations within and between are present. The programs do not behave equally nice in the following dataset, which is substantially smaller.

Table 4b: VOCAL analysis with TYPE as macro variable

	GENMOD	HLM	ML2 (R)	ML2 (NR)	VARCL
FixedPart					
(gamma)00	1.5041	1.5040	1.5040	1.5020	1.5026
(gamma)10	-.0981	-.0981	-.0981	-.0981	-.0981
(gamma)01	-.3728	-.3728	-.3723	-.3723	-.3722
(gamma)20	-.1221	-.1221	-.1221	-.1221	-.1219
(gamma)21	-.8340	-.8340	-.8350	-.8350	-.8360
RandomPart					
(sigma)	0.4016	0.4016	0.4016	0.4015	0.5016
(omega)00	0.3591	0.3590	0.3590	0.3480	0.1650
(omega)20	0.0899	0.0899	0.0899	0.0869	0.0870
Additional					
TotalIter	8	11	6	6	14
TimeIter	81	10.2	454	256	8.8
Deviance		10632			10599

3.2.4 WEBB

Individuals are 96 students (grades 7 and 8) from three average-ability LA Junior High Schools. They are in 35 small groups. The example is interesting, because the number of groups is relatively large, and thus the number of individuals per group is small. Individual level variables are posttest (POST), which is the dependent variable, pretest (PRE), and an interaction variable (NOA: asking a question and not getting an answer). The group level variable is the pretest mean in the group (PREM). The model is

$$(\underline{POST})_i = \beta_{0j} + \beta_{1j}(PRE)_i + \beta_{2j}(NOA)_i + \varepsilon_i,$$

and the between-group model is

$$\beta_{0j} = \gamma_{00} + \gamma_{01}(PREM)_j + \delta_{0j},$$

$$\beta_{1j} = \gamma_{10} + \gamma_{11}(PREM)_j + \delta_{1j},$$

$$\beta_{2j} = \gamma_{20} + \gamma_{21}(PREM)_j + \delta_{2j}.$$

After some preliminary exploration we decided on a model with both $\delta_{1j} = 0$ (coefficient of PRE is non-random) and $\gamma_{11} = 0$ (no effect of pretest mean on pretest slope). Thus the single-equation specification of the model is

$$(\underline{POST})_i = \gamma_{00} + \gamma_{01}(PREM)_j + \gamma_{10}(PRE)_i + \gamma_{20}(NOA)_i + \gamma_{21}((NOA)_i(PREM)_j)$$
$$+ (\delta_{2j}(NOA)_i + \delta_{0j} + \varepsilon_i).$$

Table 5a: WEBB data, random slope model, no macro variables

	GENMOD**	HLM*	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00		19.6895	20.5900	20.5500	20.3772
(gamma)01		-1.2115	-1.2050	-1.2220	-1.1761
(gamma)10		3.3030	3.2150	3.2220	3.2191
(gamma)20		-3.9235	-4.1170	-4.0640	-4.1021
RandomPart					
(sigma)		26.2841	26.2400	25.6100	25.8873
(omega)00		40.8725	44.5400	42.7800	2925.1110**
(omega)20		4.0382	4.6570	4.4000	3.7100
Additional					
TotaIter		200	17	26	13
TimeIter		4.3	8.2	5.8	4.3
deviance		620.2			612.0

* Stopped at 200 iterations.
** GENMOD does not work because of singularities

Table 5b: WEBB data, random slope model, PREM macro variable

	GENMOD**	HLM*	ML2(R)	ML2(F)	VARCL
FixedPart					
(gamma)00		10.9277	11.6600	11.3500	12.0932
(gamma)01		0.2545	0.0764	0.0921	0.0171
(gamma)10		3.2786	3.3420	3.3540	3.3428
(gamma)20		0.1677	0.1204	0.2575	-0.0959
(gamma)21		-0.6824	-0.6723	-0.6823	-0.6423
RandomPart					
(sigma)		25.8800	26.1700	26.3300	25.7877
(omega)00		43.4384	46.1800	43.8600	3000.8270**
(omega)20		4.4773	4.8680	4.4580	3.8280
Additional					
TotaIter		200	26	168	13
TimeIter		5.3	9.5	6.4	4.7
deviance		619.9			611.2

* Stopped at 200 iterations
** GENMOD does not work because of singularitie

In this example of group interaction within school classes the number of observations per group is very small. The number of variables outnumbers the number of observations per group. Traditional models that use LS estimation cannot estimate parameters within groups uniquely for this example. GENMOD did not run either and gave an error message about singular matrices.

3.3 Conclusions.

It is difficult to summarize the results of our numerical comparisons, but we shall try to give the main conclusions. Some of them are trivial, such as the conclusion that some computers are faster than others, and the conclusion that writing directly to the screen or a disk file is faster than sending data over telephone lines. But there are a number of conclusions that appear to be quite useful.

EM algorithms for (co)variance component analysis are nice, because they are relatively simple to program (especially in array oriented interpreted languages such as APL, MATHLAB, GAUSS), because they give monotone convergence, and because they always stay within the boundaries of the parameter space. Their convergence can be tediously slow, especially for more complicated models. HLM uses Aitkin acceleration, which makes quite a large difference, and we have reason to believe that even better acceleration algorithms are possible (Jamshidian and Jennrich, 1990). The convergence of GENMOD is sometimes intolerably slow. We are investigating the possibility of a bug that causes this, although it is partly due to the very strict convergence criteria in GENMOD. The scoring algorithm of VARCL will tend to give much faster convergence, although sometimes various parameters of the process have to be adjusted because of singularity, boundary conditions, negative eigenvalues, and divergence. We have observed sublinear convergence of VARCL in various examples, probably due to excessive damping of the upgrade. Using the results of Lindstrom and Bates (1988) could very well result in a more robust implementation of the scoring algorithm. ML 2 has an algorithm that is still somewhat of a mystery to us, but it works quite well in almost all cases. Because all the data have to be kept in core, it cannot analyze really large examples. We think that this a high price to pay for the relatively small gain in additional generality.

In general it follows from our analysis, we think, that two-level models with random slopes have very complicated likelihood surfaces. Thus maximizing the likelihood is inherently a difficult problem, unless the model is approximately true and sample size is really large (in that case OLS will give very good starting values). We also think there is much room for exploring alternatives to ML, for instance weighted least squares methods such as the ones discussed by De Leeuw and Kreft (1986), which are possibly somewhat more robust, or MINQUE/MINVE methods (Rao and Kleffe, 1989). Investigators (if the past is any indication) will tend to choose models which are too complicated (five levels, with 10 variables on each level). This leads to impossibly difficult search problems over the space of models, and to impossibly difficult likelihood maximization problems. None of the programs reviewed here can handle such problems gracefully, and something would be wrong if they could.

We have found various bugs in the programs. Some of them have been corrected, some of them (such as the value of the likelihood function in GENMOD, or the test for singularity in HLM, are still there), but none of them are very serious. All four programs tend to converge to the same solutions, which is rather nice, although there are some unpleasant exceptions.

In our comparisons we have not addressed the question about the usefulness of the statistical information. Are the likelihood ratios close to chi-squares, how accurate are the standard errors, do the estimates really improve the mean square error of OLS and WLS estimates, and so on. Such questions are important, in fact more important than computational speed or a friendly

interface, but they require more complicated research. Once you know that hierarchies exist, you see them everywhere. Thus the applicability of the software seems almost unlimited. We think that it is time to start sampling, resampling, and cross validation studies to get a more realistic idea about the possibilities of the techniques.

5. REFERENCES

Aitkin, M.A. and Longford, N. (1986). Statistical modelling issues in school effectiveness studies, **Journal of the Royal Statistical Society**, 149A, 1-43.

Bock, R.D. (ed.) (1988) **Multilevel analysis of educational data**. San Diego, Academic Press.

Burstein, L. (1980). **The analysis of multilevel data in educational research and evaluation**. In D. Berliner (ed.), Review of Research in Education, volume 8, Washington, AERA.

Bryk, A.S., Raudenbush, S.W., Seltzer M., and Congdon, R.T. (1988). **An Introduction to HLM: Computer Program and Users' Guide**. University of Chicago.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977) Maximum likelihood for incomplete data via the EM algorithm. **Journal of the Royal Statistical Society**, B39, 1-38.

Dempster, A.P., Rubin, D.B., and Tsutakawa, R.K. (1981). Estimation in covariance components models. **Journal of the American Statistical Association**, 76, 341-353.

De Leeuw, J. and Kreft, I. (1986). Random coefficient models for Multilevel Analysis. **Journal of Educational Statistics** 11, 57-85.

Goldstein, H. (1986). Multilevel mixed linear model analyssi using iterative generalized least squares. **Biometrika**, 73, 43-56.

Goldstein, H. (1987). **Multilevel models in educational and social research**. London, Griffin.

Goldstein H. (1989). Restricted unbiased iterative generalized least-squares estimation. Biometrika, 76, 3, 622-623.

Horng, S.C. (1987). Examples of sublinear convergence of the EM algorithm. **Proceedings of the Statistical Computing Section**, ASA, 266-271.

Hox,J.J., Kreft, I and Hermkens, P.L.J. (1989). **Factorial surveys; an example of multilevel design**. Amsterdam, Methodenleer en Statistiek, MLS-publicatie #40.

Humak, K.M.S. (1984). **Statistische Methoden der Modellbildung**. Volume III. Berlin, Akademie Verlag.

Jennrich, R.I. and Schluchter, M.D. (1986). Unbalanced repeated-measures models with structured covariance matrices. **Biometrics** 42, 805-820.

Jamshidian, M and Jennrich, R.I. (1990). **Conjugate gradient acceleration of the EM algorithm**. UCLA Statistics Series, #48.

Kim, K.-S. and Kreft, I. (1989). A version of HLM written in GAUSS. Internal Publication.

Kreft, I. (1987). **Methods and Models for the Measurement of School effects**. Dissertation, University of Amsterdam.

Kreft, I. and De Leeuw, J. (1989). Model based rankings of schools. UCLA Statistic Series #7. International Journal of Education, in press.

Kreft, I ,de Leeuw, J and Kim, K.S. (1990) **Comparing four different packages for hierarchical linear regression GENMOD, HLM, ML2 and VARCL**. CSE Report #110 , Center for the Study of Evaluation, UCLA.

Lindstrom, M.J. and Bates, D.M. (1988). Newton-Raphson and EM algorithms for linear mixed-effects models for replicated measures data. **Journal of the American Statistical Association**, 83, 1014-1022.

Lockheed, M.E. and Longford, N.T. (1989). **A multilevel model of school effectiveness in a developing country**. The International Bank for Reconstruction and Development/ The World Bank. Unpublished manuscript.

Longford, N.T. (1987) A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects. **Biometrika**, 74, 817-827.

Longford, N.T. (1988). **Fisher scoring algorithm for variance component analysis of data with multilevel structure**. In Bock (ed.).

Longford, N.T. (1989). Contextual effects and group-means. **Multilevel modelling newsletter**, in press.

Longford, N.T. (1989). To center or not to center. **Multilevel modelling newsletter**,3, 1989.

Mason, W. M. , Wong, G. Y. and Entwisle, B. (1984). Contextual analysis through the multilevel linear model. **Sociological Methodology**, 72-103.

Mason, W. M., Anderson, A. F. and Hayat, N. (1988). **Manual for GENMOD**. Population Studies Center, University of Michigan.

Meilijson, I. (1989.) A fast improvement to the EM algorithm on its own terms. **Journal of the Royal Statistical Society** B51, 127-138.

Rabash, J., Prosser R. and Goldstein, H (1989). **ML2. Software for Two-level Analysis. Users' Guide.** Institute of Education, University of London.

Rao, C.R. and Kleffe, J. (1989). **Estimation of variance components.** Amsterdam, North Holland.

Raudenbush, S.W. and Bryl, A.S. (1986). A hierarchical model for studying school effects. **Sociology of Education**, 59, 1-17.

Raudenbush S.W. and Bryk, A.S. (1987) Examining correlates of diversity. **Journal of Educational Statistics** 12, 241-269.

Raudenbush, S.W. (1989). "Centering" predictors in multilevel analysis: choices and consequences. **Multilevel modelling newsletter**, 2 and 3.

Schluchter, M.D. (1988). **BMDP5V -- Unbalanced repeated measures models with structured covariance matrices.** Technical Report #86, Los Angeles, BMDP Statistical Software

Stram, D.O., Laird, N.M., and Ware, J.H. (1986). **An algorithmic approach to the fitting of a general mixed ANOVA model appropriate in longitudinal settings.** In: Computer Science and Statistics: Proceedings of the seventeenth symposium on the interface. Amsterdam, North Holland.

Wong, G.Y. and Mason, W.M. (1985). The hierarchical logistic regression model for multilevel analysis. Extension of the hierarchical normal linear model for multilevel analysis. **Journal of the American Statistical Association**, 80, 513-524.

Wong, G.Y. and Mason, W.M. (1989). **Ethnicity, comparative analysis and generalization of the Hierarchical Normal Linear Model for Multilevel Analysis**. Research Report no 89-138. Population Studies Center, University of Michigan.